

Probepfprüfung (Lösung)  
**Informatik I (D-ITET / CBB)**

Felix Friedrich

ETH Zürich, 5.12.2018.

Name, Vorname: .....

Legi-Nummer: .....

Ich bestätige mit meiner Unterschrift, dass ich diese Prüfung unter regulären Bedingungen ablegen konnte, und dass ich die allgemeinen Richtlinien gelesen und verstanden habe.

*I confirm with my signature that I was able to take this exam under regular conditions and that I have read and understood the general guidelines.*

Unterschrift:

**Allgemeine Richtlinien:**

**General guidelines:**

1. Dauer der Prüfung: 45 Minuten.
2. Erlaubte Unterlagen: Wörterbuch (für von Ihnen gesprochene Sprachen).
3. Benützen Sie einen Kugelschreiber (blau oder schwarz) und keinen Bleistift. Bitte schreiben Sie leserlich. Nur lesbare Resultate werden bewertet.
4. Lösungen sind direkt auf das Aufgabenblatt in die dafür vorgesehenen Boxen zu schreiben (und direkt darunter, falls mehr Platz benötigt wird). Ungültige Lösungen sind deutlich durchzustreichen! Korrekturen bei Multiple-Choice Aufgaben bitte unmissverständlich anbringen!
5. Störungen durch irgendjemanden oder irgendetwas melden Sie bitte sofort der Aufsichtsperson.
6. Wir sammeln die Prüfung zum Schluss ab. Bitte stellen Sie unbedingt selbst sicher, dass Ihre Prüfung von einem Assistenten eingezogen wird. Stören Sie nicht die Prüfung und lassen Sie Ihre Prüfung nicht auf dem Platz liegen. Dasselbe gilt, wenn Sie früher abgehen möchten: Bitte melden Sie sich lautlos, und wir holen Ihre Prüfung ab. Vorzeitige Abgaben sind nur bis 15 Minuten vor Prüfungsende möglich.
7. Wenn Sie zur Toilette müssen, melden Sie dies einer Aufsichtsperson durch Handzeichen.
8. Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.

- Exam duration: 45 minutes.*
- Permitted examination aids: dictionary (for languages spoken by yourself).*
- Use a pen (black or blue), not a pencil. Please write legibly. We will only consider solutions that we can read.*
- Solutions must be written directly onto the exam sheets in the provided boxes (and directly below, if more space is needed). Invalid solutions need to be clearly marked. Provide corrections to answers of multiple-choice questions without any ambiguity!*
- You must not feel disturbed by anyone or anything, let the supervisor of the exam know immediately.*
- We collect the exams at the end. Important: You must ensure that your exam has been collected by an assistant. Do not take any exam with you and do not leave your exam behind on your desk. The same applies when you want to finish early: Please contact us silently and we will collect the exam. Handing in your exam ahead of time is only possible until 15 minutes before the exam ends.*
- If you need to go to the toilet, raise your hand and wait for a supervisor.*
- We will not answer any content-related questions during the exam. Please write comments referring to the tasks on the exam sheets.*

**Mock exam / Probepfprüfung  
nicht zählend / not exam relevant**

Question:	1	2	3	4	5	6	Total
Points:	6	8	7	7	10	10	48
Score:							

## Aufgabe 1: Typen und Werte (6P)

Geben Sie fur jeden der Ausdrucke jeweils C++-Typ und Wert an. *For each of the expressions provide C++ type and value.*

/1P (a) `false && false || true && true`

Typ / *Type*

`bool`

Wert / *Value*

`true | 1`

/1P (b) `3 / 4 * 4`

Typ / *Type*

`int`

Wert / *Value*

`0`

/1P (c) `489 % 7 + 489 / 7 * 7`

Typ / *Type*

`int`

Wert / *Value*

`489`

/1P (d) `0xaffe + 0x2 == 0xb000`

Typ / *Type*

`bool`

Wert / *Value*

`true | 1`

/1P (e) `10.0 + 1 / 2`

Typ / *Type*

`double`

Wert / *Value*

`10.0`

/1P (f) `'b' + 'A' - 'B'`

Typ / *Type*

`char`

Wert / *Value*

`'a'`

## Aufgabe 2: Konstrukte (8P)

Geben Sie zu folgenden Codestücken jeweils die erzeugte Ausgabe an.

*Provide the output for each of the following pieces of code.*

(a) 

```
int* a = new int[4]{1, 2, 3, 4};
int* b = &a[0];
b += 2;
std::cout << 2 * a[0];
```

Ausgabe / *Output*: 2

/2P

(b) 

```
int s = 0;
for (int i = 0; i < 10; s += ++i){}
std::cout << s;
```

Ausgabe / *Output*: 55

/3P

(c) 

```
int val = 0x111;
while (val > 0){
    std::cout << val % 2;
    val /= 2;
}
```

Ausgabe / *Output*: 100010001

/3P

### Aufgabe 3: EBNF (7P)

Die folgende EBNF definiert sehr einfache Ausdrucke (z.B. zum Suchen in Texten). Beantworten Sie die nachfolgenden Fragen.

**Anmerkung:** Leerschlage sind im Rahmen der EBNF bedeutungslos.

*The following EBNF defines very simple expressions (e.g. to search in texts). Answer the questions below.*

**Remark:** Whitespaces are irrelevant in the context of this EBNF.

```
expression = term { term }.
term       = factor [quantor].
factor     = letter | "[" letter "-" letter "]" | "(" expression)".
quantor    = "*" | "?" | "+".
letter     = "a" | "b" | "c" | "d" | "e" | "t" | "h".
```

/5P

(a) Geben Sie in folgender Matrix an, ob die Zeichenkette ein Ausdruck (expression) gemass der EBNF entspricht oder nicht. Richtige Antworten ergeben 1 Punkt, falsche Antworten ergeben -1 Punkt.

*Provide in the following matrix if the string corresponds to a valid expression according to the EBNF. Correct answers yield 1 point, wrong answers yield -1 point.*

Zeichenkette <i>String</i>	gultig <i>valid</i>	ungultig <i>invalid</i>
[a-c]*	x	
c++		x
(abc)()?		x
a?b+c*(d?)*	x	
(eth)*	x	

/2P

(b) andern Sie genau eine Produktionsregel der EBNF ab, so dass die folgenden vier Zeichenketten (ohne Gansefusschen) jeweils gultige Expressions sind.

*Modify one and only one production rule of the EBNF such that the following four expressions (without the quotes) are valid expressions each.*

"(a|b)\*"    "abc|cde"    "a?|b\*|c+"    "(ab)\*|(cd)+"

Geanderte Zeile / *Modified line:*

```
term = factor [quantor] {"|" factor [quantor]}. or
expression = term { ["|"] term }
```

### Aufgabe 4: Fließkommasyeme (7P)

- (a) Beantworten Sie die folgende Fragen und nehmen Sie dafür an, dass Fließkommazahlen sich gemäss dem IEEE-Standard 754 für Fließkommazahlen-Arithmetik verhalten! Richtige Antworten ergeben 1 Punkt, falsche Antworten ergeben -1 Punkt.

*Answer the following questions, assuming that floating point numbers behave according to the IEEE 754 standard for floating point arithmetics! Correct answers yield 1 point, wrong answers yield -1 point.*

/3P

Frage Question	wahr <i>true</i>	falsch <i>false</i>
Die Literale 0.1 und 0.1f haben gleich viele signifikante Stellen. <i>Literals 0.1 and 0.1f have the same number of significant digits.</i>		x
Es gibt ein Wert a vom Typ float für den gilt: $a + 1 == a$ . <i>There exists a value a of type float for that holds: <math>a + 1 == a</math>.</i>	x	
Jede 32-bit Zahl vom Typ int kann ohne Wertänderung in den Typ float konvertiert werden. <i>Each 32-bit number of type int is convertible into the type float without change in value.</i>		x

- (b) Beantworten Sie die folgende Fragen zum normalisierten Fließkommasytem  $F^*(\beta, p, e_{min}, e_{max})$  mit  $\beta = 2, p = 5, e_{min} = -3, e_{max} = 3$ . Richtige Antworten ergeben 1 Punkt, falsche Antworten ergeben -1 Punkt.

*Answer the following questions regarding the normalized floating point system  $F^*(\beta, p, e_{min}, e_{max})$  with  $\beta = 2, p = 5, e_{min} = -3, e_{max} = 3$ . Correct answers yield 1 point, wrong answers yield -1 point.*

/4P

Frage Question	wahr <i>true</i>	falsch <i>false</i>
Das Fließkommasytem $F^*$ kann die Zahl 0 repräsentieren. <i>The floating point system <math>F^*</math> can represent the number 0.</i>		x
15 ist die grösste Zahl, welche im Fließkommasytem $F^*$ repräsentierbar ist. <i>15 is the biggest number representable in floating point system <math>F^*</math></i>		x
1/16 ist im Fließkommasytem $F^*$ exakt repräsentierbar. <i>1/16 is exactly representable in the floating point system <math>F^*</math></i>		x
11 und 11.5 folgen im Fließkommasytem $F^*$ direkt aufeinander, d.h., es gibt keine Zahl $Z \in F^*$ für die gilt $Z > 11$ und $Z < 11.5$ . <i>In <math>F^*</math>, 11 and 11.5 follow each other without a gap, i.e., there is no number <math>Z \in F^*</math>, where <math>Z &gt; 11</math> and <math>Z &lt; 11.5</math>.</i>	x	

## Aufgabe 5: Rekursion (Pascalsches Dreieck) (10P)

Im folgenden ist das **Pascalsche Dreieck**  $p$  (in einer Matrixvariante) abgebildet.

Der Eintrag  $p(1, 1)$  in Zeile  $r = 1$  und Spalte  $c = 1$  hat den Wert 1:  $p(1, 1) = 1$ .

Darüber hinaus gilt  $p(r, c) = 0$  für alle  $c \leq 0$  oder für alle  $r \leq 0$ .

Jeder andere Wert an Zeile  $r$  und Spalte  $c$  wird durch Addition der Zahlen darüber (Zeile  $r-1$ , Spalte  $c$ ) und links davon (Zeile  $r$ , Spalte  $c-1$ ) gebildet.

*Below Pascal's Triangle  $p$  is shown (in a matrix variant).*

*The number  $p(1, 1)$  in row  $r = 1$  and column  $c = 1$  has value 1:  $p(1, 1) = 1$ .*

*Moreover it holds that  $p(r, c) = 0$  for all  $c \leq 0$  or for all  $r \leq 0$ .*

*Every other value at row  $r$  and column  $c$  is provided as the value above, at row  $r-1$  and column  $c$ , and left, row  $r$  and column  $c-1$ .*

	Spalten / columns →					
Index	0	1	2	3	4	...
0	0	0	0	0	0	...
1	0	1	1	1	1	...
2	0	1	2	3	4	...
3	0	1	3	6	10	...
4	0	1	4	10	20	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Zeilen / rows ↓



- (a) Ergänzen Sie die folgende Funktion so, dass sie die Zahl  $p(r, c)$  in Zeile  $r$  und Spalte  $c$  berechnet.

*Complement the following method such that it computes the number  $p(r, c)$  in row  $r$  and column  $c$ .*

/6P

```
// Compute number in Pascal's triangle at row r and column c
int p(int r, int c){
    if (r <= 0 || c <= 0)
        [1p]
        return 0;
}
if (c == 1 && r == 1) [or also works]
    [1p]
    return 1;
}
return p(r-1,c) + p(r,c-1);
[4p]
}
```

- (b) Geben Sie einen kurzen informellen Beweis dafür an, dass Ihre Methode für alle Eingaben (Parameter)  $r \geq 0, c \geq 0$  terminiert.

*Provide a short informal proof that your method will terminate for all inputs (parameters)  $r \geq 0, c \geq 0$ .*

/4P

For each recursive call of  $p$  the  $r$  or  $c$  argument decrease. When  $r$  or  $c$  reach 0 the base case is reached and no further call takes place. [2p for decrease, 2p for mentioning the base case. 4p for any other correct complete explanation]

## Aufgabe 6: Mengen-Datentyp (10P)

Die Klasse Set implementiert einen Datentyp zur Representation von Mengen. Elemente von 0 bis 9 konnen in einem Set gespeichert werden. Beantworten Sie die nachfolgende Fragen.

*The class Set implements a datatype to represent a set. The set can contain elements from 0 to 9. Answer the following questions.*

```
class Set {
public:
    // POST: Empty set.
    Set() { for(int i = 0; i < 10; ++i) elems[i] = false; }
    // POST: Adds element to set.
    void add(int nr) { elems[nr] = true; }
    //POST: Removes element from set.
    void remove(int nr) { elems[nr] = false; }
    // POST: Returns the union of set this and set other.
    Set operator+(const Set& other) const;
    // POST: Returns true if set this and set are equal, otherwise false.
    bool operator==(const Set& other) const;
    // POST: Returns the intersection of set this and set other.
    Set operator*(const Set& other) const;

    bool elems[10];
}
```

- /4P (a) Implementieren Sie den Operator *Implement the operator* `Set::operator*`, welcher die Schnittmenge zweier Mengen zuruckgibt. *Set::operator\* that returns the intersection of two sets.*

```
Set operator*(const Set& other) const {
    Set set;
    for(int i = 0; i < 10; ++i) {
        set.elems[i] = elems[i] && other.elems[i];
    }
    return set;
}
```



- (b) Wie kann man die Kapselung der Klasse Set verbessern, ohne Funktionalitat einzubussen? (Maximal 2 sehr kurze Satze!)

*How can the encapsulation of class @Set@ be improved without losing functionality? (Maximally 2 very short sentences!)*

/2P

Make access to field elems private. Provide getter and setter functions.

- (c) Fur diese Teilaufgabe durfen Sie vom Set Datentyp ausschliesslich die Mitgliedsfunktionen verwenden!

Geben Sie fur jeden der folgenden Vergleichsoperatoren den entsprechenden Ausdruck an, so dass die angegebene Nachbedingungen der Operatoren jeweils erfullt sind.

**For this sub task, from data type Set you may only use its member functions!**

*For each of the following comparison operators specify the expression so that the given postcondition of each operator is fulfilled.*

/4P

```
// POST: Returns true if set a and set b are unequal, otherwise false.
bool operator!=(const Set& a, const Set& b) const {
    return !(a == b);
}
// POST: Returns true if set a is a subset of set b.
bool operator<=(const Set& a, const Set& b) const {
    return a * b == a;
}
// POST: Returns true if set a is a superset of set b.
bool operator>=(const Set& a, const Set& b) const {
    return a + b == a;
}
// POST: Returns true if set a is a proper (strict)
// subset of set b.
bool operator<(const Set& a, const Set& b) const {
    return a * b == a && !(a == b);
}
```